

Remarks

Claims 1-30, 32-33 and 36-48 are currently pending in this application. Claims 41-48 are new. Claims 1, 3-6, 11-14, 17, 20, 22-31, and 33 are currently amended. Claims 31, 34 and 35 are cancelled in this amendment.

Claims 1-5, 7-10, 13-20 and 22-40 were rejected under 35 U.S.C. 102(b) as being anticipated by “Managing Long Linked Lists Using Lock-Free Techniques” by FAROOK et al. The Applicant traverses these rejections.

Regarding Claim 1,

Amended Claim 1 recites:

1. *A method for executing an operation upon a linked data structure having at least one element, the method comprising the steps of:*
 - (d) *performing a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions;*
 - (e) *developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions and being associated with a set of pointers to the linked data structure, the set of pointers being stored external to the linked data structure, the second set of element state transitions being distinct from the first set of element state transitions; and*
 - (a) *performing the second set of operation tasks in a second phase using the set of pointers.*

Claim 1, as amended, includes the limitations “the set of pointers being stored external to the linked data structure.” The Applicant is unable to find these and other added limitations in the cited art and believes that Claim 1 is, therefore, allowable for at least these reasons.

The amendments to Claim 1 are supported by the application as filed, for example, in paragraph [0024] and FIG. 2.

Regarding Claim 2,

The Applicant believes that Claim 2 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

Regarding Claim 3,

Amended Claim 3 recites:

- 5 3. *(Currently Amended) The method of claim 1 wherein the step of developing a second set of operation tasks further comprises developing the set of pointers to the data structure.*

Claim 3, as amended, includes the limitations “*developing a second set of operation tasks ... comprises developing the set of pointers [stored external to the linked data*
10 *structure].*” The Applicant is unable to find any teaching of these limitations within the cited art and believes that Claim 3 is allowable for at least these reasons. The Applicant further believes that Claim 3 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

Regarding Claim 4,

15 Amended Claim 4 recites:

4. *(Currently Amended) The method of claim 1 wherein the first phase comprises performing parallel operations on the linked data structure and the second phase comprises performing serial operations on the linked data structure, each of the serial operations being developed during one of the parallel operations..*

20 Claim 4, as amended, includes the limitations specifying that the first phase includes performing “*parallel operations*” and the second phase includes performing “*serial operations*” “*developed during one of the parallel operations.*” The Applicant is unable to identify teaching of these limitations in the cited art. The Applicant believes that Claim 4 is
25 allowable for at least these reasons. The Applicant further believes that Claim 4 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

Regarding Claim 5,

Amended Claim 5 recites:

5. (Currently Amended) *The method of claim 1 wherein the set of pointers are stored in a list.*

5 Claim 5 is amended to include the limitations that the “*set of pointers [being stored external to the linked data structure] are stored in a list.*” The Applicant is unable to identify any teaching within the cited art of a list of pointers stored external to the linked data structure, much less that this set of pointers are used for “*performing the second set of operation tasks,*” as recited in Claims 1 and 5. The Applicant believes that Claim 5 is
10 allowable for at least these reasons. The Applicant further believes that Claim 5 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

Regarding Claim 7,

Claim 7 recites:

7. *The method of claim 1 wherein the step of developing a second set of operation tasks further comprises a step of performing a conflicts check for the operation.*
15

In rejecting Claim 7, the Examiner states “FAROOK teaches a step of performing a conflicts check for the operation (via if the node is being accessed by another process) (via the try_* operations) (pg. 11-13).” Assuming for the sake of argument that the cited text
20 taught a conflicts check, the Applicant respectfully points out that this hypothetical conflicts check would be performed during execution of those operations suggested by the Examiner as teaching the second set of operation tasks. For example, the last sentence on page 6 of FAROOK teaches checking a version number during a deletion. The Applicant believes that the Examiner is suggesting that this text teaches “*performing a conflicts check*” and points
25 out that the version number check occurs during the last steps and, thus, the suggested “second phase” of the deletion. In contrast, Claim 7 includes limitations that specify that a

conflict check is performed during “*the step of developing*,” rather than during the step of “*performing the second set of operation tasks*.” For at least these reasons, the Applicant believes that Claim 7 is allowable. The Applicant further believes that Claim 7 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

5 **Regarding Claim 8,**

Claim 8 recites:

8. *The method of claim 1 wherein the first set of element state transitions further comprises:*

- 10 (a) *a valid state to a pending delete state transition;*
 (b) *a pre-associated state to a pending insert state transition; and*
 (c) *a pending insert state to a hidden state transition.*

In rejecting Claim 8, the Examiner states that the limitations of Claim 8 are taught by

FAROOK:

15 (via performing the cursor function in order to get to the correct insert or delete
 positions for performing the final state, actually insertion or deletion into/from the
 list((pg. 5-6, 2.3 Non-Blocking Linked list wherein a prior teaching discloses the
 limitation and on page 12-14 and 11-12, wherein the enhancement to the prior
20 teachings teach the limitations.)

The Applicant traverses this statement.

First, it is unclear to the Applicant how the cited text teaches the limitations of Claim

8. For example, “performing the cursor function in order to get to the correct insert or delete
positions” does not appear to alter the state of an element. Thus, it cannot teach any of the
25 state transitions recited in Claim 8. Further, actually inserting and deleting an element in a
list does not necessarily include any “*pending*” or “*hidden*” states to transition between as
recited in Claim 8. The Applicant, therefore, respectfully requests that the Examiner
specifically point out each of the state transitions recited in Claim 8, or allow Claim 8.

Second, it is unclear to the Applicant which teachings of the prior art the Examiner believes teaches “a valid state,” “a pending delete state,” “a pre-associated state,” “a pending insert state,” and “a hidden state,” as recited in Claim 8. For example, if for the sake of argument one were to assume that the state of the “new” element shown in Figure 3 of FAROOK taught a “pending insert state,” then it is unclear to the Applicant what teaching of FAROOK could teach the “a pre-associated state” or “a hidden state.” The Applicant, therefore, requests that the Examiner specifically point out those teachings of FAROOK thought to teach each of these states, or allow Claim 8.

Third, Claim 8 recites a set of state transitions that are included in “the first set of element state transitions,” which are performed in “a first phase” according to Claim 1. Thus, “a valid state to a pending delete state transition” and “a pending insert state to a hidden state transition,” are both performed in a first phase. These state transitions involve both insertion and deletion. The first of these state transitions is part of a deletion operation and the second of these state transitions is part of an insertion operation. The Applicant is unable to find any teaching within the cited art that operation tasks of a deletion operation and operation tasks of an insertion operation are performed as part of a first phase. The Applicant, therefore, requests that the Examiner point out, with specificity, teaching of “a valid state to a pending delete state transition” and “a pending insert state to a hidden state transition,” within “a first phase,” or allow Claim 8.

The Applicant further believes that Claim 8 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

Regarding Claim 9,

Claim 9 recites:

9. The method of claim 1 wherein the second set of element state transitions further comprises:

- (a) a pending insert state to a valid state transition;
- (b) a pending delete state to an invalid state transition;
- 5 (c) a hidden state to an invalid state transition;
- (d) a pending delete state to a valid state transition;
- (e) a hidden state to a pending insert state transition; and
- (f) a pending insert state to an invalid state transition.

10 The Applicant believes that Claim 9 is allowable for reasons similar to those discussed in relation to Claim 8. Specifically, the teachings cited by the Examiner do not appear to teach the particular element state transitions recited in Claim 9, *"in a second phase."* For example, the "TryDelete" or "TryInsert" functions cited by the Examiner do not appear to necessarily teach any of the recited state transitions. It is possible to attempt a
15 deletion or insertion without performing all of the transitions recited as limitations in Claim 9.

Further, as discussed in relation to Claim 8, it is unclear to the Applicant which teachings of the prior art the Examiner believes teaches *"a valid state," "a pending delete state," "a pre-associated state," "a pending insert state,"* and *"a hidden state,"* etc. as recited
20 in Claim 9. The Applicant requests that the Examiner specifically point out teaching of each of these states, or allow Claim 9.

Further, as discussed in relation to Claim 8, the recited state transitions are performed in the same *"first phase."* Even if, for the sake of argument, the TryDelete and TryInsert functions of the cited art taught one of the state transitions of Claim 9, there does not appear
25 to be any teaching within FAROOK that the TryDelete and TryInsert functions are performed in a same phase. The Applicant requests that the Examiner specifically point out such teaching, or allow Claim 9.

The Applicant further believes that Claim 9 is allowable for at least the reasons discussed with respect to Claim 1, from which it depends.

Regarding Claim 10,

Claim 10 recites:

- 5 10. *A method for performing insertion and deletion operations on elements in a linked data structure, the method comprising the steps of:*
- (a) *performing a first set of operation tasks in a first phase for each insertion and deletion operation, the first set of operation tasks operable to effect a first set of element state transitions;*
- 10 (b) *developing a second set of operation tasks for each insertion and deletion operation, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and*
- (c) *performing the second set of operation tasks in a second phase.*

15 In rejecting Claim 10, the Examiner suggests that the limitations of “(a) *performing a first set of operation tasks in a first phase*” are taught “(via performing the cursor operation to find the correct position in the list for insertion or deletion / via creating the new node, pointing its next identifier to the c.target and checking if c.target has been changed)...” The

20 Applicant traverses this statement.

 Claim 10 recites performing both “*insertion and deletion operations.*” In particular, the first step includes “(a) *performing a first set of operation tasks in a first phase for each insertion and deletion operation.*” Thus, the first phase includes operation tasks for both deletion and insertion operations. The Applicant is unable to identify any teaching within the

25 cited art of performing operation tasks for both deletion and insertion operations in a first phase. Even if one were to assume for the sake of argument that the deletion operation of FAROOK was divided into an initial phase and a secondary phase, and that the insertion operation of FAROOK was divided into an initial phase and a secondary phase, there is no teaching in FAROOK that the phases of deletion are in any way related to the phases of

insertion. Thus, any deletion and insertion operations taught in FAROOK appear to be totally independent of each other. In contrast, Claim 10 recites “a first phase” in which “operation tasks ... for each insertion and deletion operation” are performed. These limitations are not taught by independent deletion and insertion operations.

5 Further, Claim 10 recites “performing the second set of operation tasks [for each insertion and deletion operation] in a second phase,” which is similarly not taught by the cited art. The Applicant, therefore, requests that the Examiner specifically point out what is thought to be the “first phase” and “second phase” in the cited art, and a teaching that operation tasks of the insertion and deletion tasks are performed in each of these phases, or
10 allow Claim 10.

Regarding Claim 13,

Amended Claim 13 recites:

13. (Currently Amended) A method of inserting a plurality of elements into a linked data structure comprising the steps of:

- 15 (a) performing a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions including a pre-associated state to a pending insert state transition for each of the plurality of elements;
- 20 (b) developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions including a pending insert state to a valid state transition for each of the plurality of elements; and
- (c) performing the second set of operation tasks in a second phase.

Claim 13, as amended, is directed toward “inserting a plurality of elements into a
25 linked data structure,” and includes limitations that specify that “a first set of operation tasks ... for each of the plurality of elements” are performed “in a first phase,” and “a second set of operation tasks ... for each of the plurality of elements” are performed “in a second phase.”

While the cited art teaches insertion of elements into a data structure, there does not appear to be any teaching that “*inserting a plurality of elements*” are performed together in a first phase and that other parts of the same insertions are then performed together in a second phase.

The Applicant, therefore, requests that the Examiner specifically point out how the cited art teaches all the limitations recited in Claim 13, or allow Claim 13. Specifically, the Applicant requests that the Examiner point out how the cited art teaches “*performing ... a first set of element transitions ... for each of the plurality of elements*” “*in a first phase,*” and “*performing the second set of operations [operable to effect a second set of element state transitions ... for each of the plurality of elements] in a second phase.*”

Regarding Claim 14,

The Applicant believes that Claim 14 is allowable for at least the reasons discussed with respect to Claim 13, from which it depends.

Regarding Claim 15,

The Applicant further believes that Claim 15 is allowable for at least the reasons discussed with respect to Claims 13 and 14, from which it depends.

Regarding Claim 16,

Claim 16 recites:

The Applicant further believes that Claim 16 is allowable for at least the reasons discussed with respect to Claims 13, 14 and 15, from which it depends.

Regarding Claim 17,

Amended Claim 17 recites:

17. *(Currently Amended) A method of deleting a plurality of elements from a linked data structure comprising the steps of:*

- 5 (a) *performing a first set of operation tasks in a first phase, the first phase including a valid state to a pending delete state transition for each of the plurality of elements;*
- (b) *developing a second set of operation tasks, for execution in a second phase including a pending delete state to an invalid state transition for each of the*
10 *plurality of elements; and*
- (c) *performing the second set of operation tasks in the second phase.*

Claim 17 is amended so as to be directed to a “*method of deleting a plurality of elements from a linked data structure,*” and to include limitations specifying that “*a valid*
15 *state to a pending delete state transition for each of the plurality of elements*” are performed together in “*a first phase*” and “*a pending delete state to an invalid state transition for each of the plurality of elements*” are performed together in “*a second phase.*”

As discussed above with respect to Claims 10 and 13, the Applicant is unable to find any teaching that insertion and deletion operations include operation tasks (of each) that are
20 performed “*in a first phase.*” As discussed above, if the insertion and deletion operations of FAROOK were to be assumed to be divided into multiple phases, these would be independent phases and parts of an insertion operation and of a deletion operation would not be executed “*in a first phase.*”

The Applicant requests that the Examiner point out teaching of all the limitations of
25 Claim 17 within the cited art, or allow Claim 17.

Regarding Claims 18-20,

The Applicant believes that Claims 18-20 are allowable for at least the reasons discussed herein with respect to Claim 17, from which they depend.

Regarding Claim 22,

Amended Claim 22 recites:

22. *(Currently Amended) A method for executing operations upon a linked data structure having at least one element, the method comprising the steps of:*

- (a) *creating first and second sets of operation tasks, the first set of operation tasks being characterized by navigation of the linked data structure using at least an existing link, and the second set of operation tasks being distinct from the first set of operation tasks and each being characterized by at least a different pointer from outside the linked data structure to the linked data structure; and*
- (b) *performing the first set of operation tasks in a first phase and the second set of operation tasks in a second phase.*

Claim 22 is amended to include the limitations that “the second set of operation tasks,” are “each being characterized by at least a different pointer from outside the linked data structure to the linked data structure.” The Applicant is unable to identify any teaching within the cited art of a list of pointers stored external to the linked data structure, much less that these different pointers characterize “*the second set of operation tasks*,” as recited in Claim 22. The Applicant believes that Claim 22 is, therefore, allowable for at least these reasons.

The Applicant further believes that Claim 22 is allowable for at least the reasons discussed with respect to Claims 1 and 5.

Regarding Claim 23,

Amended Claim 23 recites:

23. *(Currently Amended) A method for executing a plurality of operations upon a linked data structure having at least one element, the method comprising the steps of:*

- (a) *dividing each of the plurality operations into first and second distinct sets of operation tasks;*
- (b) *performing the first set of operation tasks of the plurality of operations together in a first phase; and*
- (c) *performing the second set of operation tasks of the plurality of operations together in a second phase.*

The Applicant believes that, as amended, Claim 23 is allowable for at least the reasons discussed herein with respect to Claims 10 and 13. Specifically, the cited art does not teach “performing” part of “a plurality of operations together in a first phase” and then “performing” a second part of “the plurality of operations together in a second phase.” The Applicant requests that the Examiner specifically point out these teachings within the cited art or allow Claim 23.

Regarding Claim 24,

Amended Claim 24 recites:

24. *The method of claim 23 wherein the first set of operation tasks is operable in parallel to maintain the linked data structure in an existing linked state.*

Claim 24 as amended recites “the first set of operation tasks is operable in parallel.”

The Applicant is unable to identify any teaching in the cited art that the taught operations are performed in parallel. The Applicant believes that Claim 24 is allowable for at least this reason.

The Applicant further believes that Claim 24 is allowable for at least the reasons discussed with respect to Claim 23, from which it depends.

Regarding Claim 25,

Amended Claim 25 recites:

25. *The method of claim 24 wherein the second set of operation tasks is operable in series to modify the existing linked state.*

Claim 25 is amended to include the limitations that “the second set of operation tasks is operable in series.” The Applicant is unable to identify any teaching in the cited art that the taught operations are performed in series, much less that part of an operation is performed in parallel (as recited in Claim 24) and another part of an operation is performed in series (as

recited in Claim 25). The Applicant believes that Claim 24 is allowable for at least these reasons.

The Applicant further believes that Claim 25 is allowable for at least the reasons discussed with respect to Claims 23 and 24, from which it depends.

5 **Regarding Claim 26,**

The Applicant believes that Claim 26 is allowable for at least the reasons discussed with respect to Claim 23, from which it depends.

Regarding Claim 27,

10 The Applicant believes that Claim 27 is allowable for at least the reasons discussed with respect to Claims 23 and 26, from which it depends.

Regarding Claim 28,

Amended Claim 28 recites,

- 15 28. *(Currently Amended) A system for executing an operation upon a linked data structure having at least one element, the system comprising:*
- (a) *a memory for storing the linked data structure;*
 - (b) *a processor coupled to the memory, the processor operable to perform a first set of operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions, to develop a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions, and to perform the second set of operation tasks in a second phase by using a plurality of pointers stored external to the linked data structure to navigate to a plurality of nodes within the linked data structure.*
- 20
- 25

25 Claim 28 as amended includes the limitations “to perform the second set of operation tasks in a second phase by using a plurality of pointers stored external to the linked data structure to navigate to a plurality of nodes within the linked data structure.” The cited art does not appear to include teachings of operation tasks that are performed “without

30 *navigating between elements of the linked data structure.*” For example, in the cited art, all

steps of the insertion operation illustrated by Figure 3 require navigation between the c.pre_aux element and the new element, between new element and the new.aux element, or between the new.aux element and the c.target element. For at least these reasons, the Applicant believes that Claim 28 is allowable.

5 **Regarding Claim 29,**

Amended Claim 29 recites:

29. *(Currently Amended) A system for executing an operation upon a linked data structure having at least one element, the system comprising:*

- 10 (a) *a first memory space for storing the linked data structure;*
- (b) *a second memory space for storing a plurality of pointers separate from the linked data structure; and*
- (c) *a processor coupled to the first and second memory and operable to divide the operation into first and second distinct sets of operation tasks, perform the first set of operation tasks in a first phase,*
15 *develop a second set of operation tasks in the first phase, development of the second set of operation tasks including storing at least one pointer to the linked data structure in the second memory space, and*
perform the second set of operation tasks in a second phase using the at least one pointer.

20 Claim 29 as amended recites a variety of limitations including “a second memory space for storing a plurality of pointers separate from the linked data structure,” “development of the second set of operation tasks including storing at least one pointer to the linked data structure in the second memory space,” and “perform the second set of operation
25 tasks in a second phase using the at least one pointer.” The Applicant believes that these limitations are not taught by the cited art. For example, FAROOK appears not to teach a memory space that is separate from the linked data structure and use of pointers in that memory to complete a second phase of operation tasks. For at least these reasons, the Applicant believes that Claim 29 is allowable.

30 **Regarding Claim 30,**

Amended Claim 30 recites:

30. *(Currently Amended) A computer readable medium for executing a plurality of operations upon a linked data structure having at least one element, the computer readable medium comprising:*

- 5 (a) *a code segment for performing a plurality of first phase operation tasks in a first phase, each of the plurality of operations being associated with a different member of the first phase operation tasks;;*
- (b) *a code segment for developing a plurality of second phase operation tasks during the first phase, , each of the plurality of operations being associated*
10 *with a different member of the plurality of second phase operations;*
- (c) *a code segment for performing the plurality of second phase operations tasks in a second phase after completion of the first phase.*

 The Applicant believes that Claim 30 is allowable for the same reasons discussed
15 above with respect to Claims 10, 13, and 23. Specifically, the cited art does not teach the execution of operation tasks of a plurality of operations in a first phase.

Regarding Claim 32,

 The Applicant believes that Claim 32 is allowable for at least the same reasons discussed above with respect to Claim 1.

20 **Regarding Claim 33,**

 The Applicant believes that Claim 33 is allowable for at least the same reasons discussed above with respect to Claim 11.

Regarding Claim 36,

Claim 36 recites:

- 25 36. *A consistent method of executing simultaneous operations on a linked data structure having at least one element, the method comprising the steps of:*
performing any first phase operation task of each of the simultaneous operations in a first phase using parallel processes;
developing a set of serial operations during the first phase; and
30 *performing any second phase operation task of each of the simultaneous operations in a second phase, the second phase operation task including at least one of the set of serial operations.*

The Applicant believes that Claim 36 is allowable for at least the reasons discussed with respect to Claims 10, 13, 23 and 30.

In rejection of Claim 36, the Examiner suggests that “(processes that are trying to performing operations on adjacent nodes...) teaches “*using parallel processes*,” and “(via only one process succeeds while the others fail)” teaches “*a set of serial operations*.” The Applicant traverses this statement.

A teaching that “one process succeeds while others fail” does not teach “*a set of serial operations*,” as recited in Claim 36. The term “serial” refers to “of or relating to the sequential performance of multiple operations,” (The American Heritage® Dictionary of the English Language, Fourth Edition, by Houghton Mifflin Company). The fact that one operation fails while the others may succeed to not make the operations serial if they are being attempted at the same time. Operations become parallel once their execution overlaps in time and do not turn from parallel to serial if one fails. The cited art, therefore, does not teach “*developing a set of serial operations during the first phase*,” and in particular where the first phase uses “*parallel processes*.” The Applicant believes that Claim 36 is allowable for at least these reasons.

The Applicant further believes that Claim 36 is allowable for the same reasons discussed above with respect to Claims 10, 13, and 23. Specifically, the cited art does not teach the execution of operation tasks of a plurality of *simultaneous* operations in a first phase.

Regarding Claims 37 and 39,

The Applicant believes that Claims 37 and 39 are allowable for at least the reasons discussed with respect to Claim 36, from which they depend.

Regarding Claim 38,

Claim 38 recites:

38. *(Original) The method of claim 36 wherein at least one of the simultaneous operations includes an element deletion operation, the second phase operation task of the element deletion operation being performed independently of navigation of the linked data structure.*

In rejecting Claim 38, the Examiner suggests that “(delete operation) (pg. 12 – 13) (wherein one can still traverse the list based on the traverse pointer) (pg. 9)” teaches “*the*

element deletion operation being performed independently of navigation of the linked data structure,” as recited in Claim 38. The Applicant traverses this statement. All operation tasks of FAROOK appear to require navigation between nodes of the linked data structure.

For example, in the cited art, all steps of the insertion operation illustrated by Figure 3 requires navigation between the c.pre_aux element and the new element, between new

element and the new.aux element, or between the new.aux element and the c.target element.

For eat least these reasons, the Applicant believes that the art cited by the Examiner does not teach the limitations of Claim 38 and that Claim 38 is allowable.

Regarding Claim 40,

Claim 40 recites:

40. *(Original) The method of claim 36 wherein the first phase operation tasks of more than one of the simultaneous operations are completed before the second phase of any of the simultaneous operations is initiated.*

In rejecting Claim 40 the Examiner states “FAROOK teaches the first phase operation tasks of more than one of the simultaneous operation are completed before the second phase of any of the simultaneous operations is initiated (via traversing the list independently of other operations) (pg. 9-13).” The Applicant traverses this statement.

It is unclear to the applicant how the teaching cited by the Examiner teaches the claim limitations as suggested by the Examiner. For example, how does traversing the list independently assure that “the first phase operation tasks are completed before the second phase of **any** of the simultaneous operations is initiated”? The Applicant is unable to find any teaching within the cited art that first phases must be completed before **any** second phases are initiated. The operations of FAROOK appear to be executed independently. Thus, a second phase of one operation could start before the first phase of another operation is completed. The Applicant, therefore, requests that the Examiner more specifically point out teachings of the limitations of Claim 40 within the cited art, or allow Claim 40.

Claims 6, 11, 12 and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Managing Long Linked Lists Using Lock-Free Techniques” by FAROOK et al.

Regarding Claim 6,

Amended Claim 6 recites:

6. (Currently Amended) The method of claim 1 wherein the set of pointers are stored in a first in last out list.

In rejecting Claim 6, the Examiner admits that Claim 6 does not teach “that the list is a first in last out list.” The Examiner then takes official notice that first in last out lists are well known in the art and that the use of such a list in the context of FAROOK would thus be obvious.

The Applicant traverses these statements of the Examiner.

First, FAROOK teaches only one list and that list is clearly not a first in last out list because elements are being inserted or deleted from the middle rather than the ends of the list

(see Figure 3). All pointers taught in FAROOK are part of, and thus stored in, the one list taught by FAROOK. Thus, contrary the Examiner's suggestion, the only storage of pointers taught in FAROOK are clearly not in a first in last out list.

5 Second, it is not clear how a first in last out list would be used for storing pointers in the context of FAROOK as suggested by the Examiner. In FAROOK pointers are stored in unique nodes as shown in Figure. 4. There would be no reason for these nodes to include first in last out lists because there is only a limited and constant number of fields in each node. Further, as discussed above, these nodes are clearly not themselves included in a first in last out list because, as shown in Figure 3, the whole purpose of FAROOK is to insert or
10 delete nodes from the middle of a linked list. The use of a first in last out list as suggested by the Examiner, thus, appears contradictory and impractical in the context of FAROOK.

The Applicant, therefore, believes that Claim 6 is allowable at least because the cited art does not teach storing a set of pointers in a first in last out list, and for the reasons discussed with respect to Claim 1, from which it depends.

15 Further, the Examiner does not appear to provide a motivation to include the use of a first in last out list within the teachings of FAROOK. As such, the Examiner has not presented a prima facie case for rejection of Claim 6 under 103(a). The Applicant respectfully requests that the Examiner provide a proper motivation from the prior art as required, or allow Claim 6.

20 **Regarding Claim 11,**

Amended Claim 11 recites:

11. *(Currently Amended) A method for executing operations upon a linked data structure having at least one element, the method comprising the steps of:*

- 25 (a) *queuing operation tasks in a task queue;*
(b) *receiving the queued operation tasks;*

- (c) performing a first set of the operation tasks in a first phase, the first set of operation tasks operable to effect a first set of element state transitions;
- (d) developing a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and
- (e) performing the second set of operation tasks in a second phase.

Regarding Claim 11, the Examiner admits “FAROOK does not teach queuing operation tasks in a task queue and receiving the queued operation tasks,” and take official notice that operations can be stored in a queue and serialized.

The Applicant traverses this statement and respectfully points out that those items that are queued in Claim 11 are “*operation tasks*” which are distinguished from “*operations*.”

Specifically, an operation task is a sub-step of an operation that is performed to accomplish

the operation. Even if for the sake of argument one were to assume that queuing of operations was commonly practiced as suggested by the Examiner, this does not mean that “*operation tasks*” are well known in the art as being queued. Thus, even when considering the official notice taken by the Examiner, the Examiner has not shown all the elements of Claim 11 and, thus, has not made a prima facie case for the rejection of Claim 11 under

103(a). In FAROOK any sub-steps of an operation appear to be always performed in the same order. There would, thus, be no reason to form a queue or for the step of “*receiving the queued operation tasks*,” as recited in Claim 11. It is not clear to the Applicant what would be the purpose of queuing “*operation tasks*” in the context of FAROOK. For at least these reasons, the Applicant believes that Claim 11 is allowable.

The Applicant further believes that Claim 11 is allowable for at least the reasons discussed with respect to Claims 1, 10, 13 and 23.

Further, the Examiner does not appear to provide a motivation to include the use of a task queue and receiving the queued tasks within the teachings of FAROOK. As such, the Examiner has not presented a prima facie case for rejection of Claim 11 under 103(a). The Applicant respectfully requests that the Examiner provide a proper motivation from the prior art as required, or allow Claim 11.

Further, the while the Examiner refers to “serialized” operations, the Applicants respectfully point out that this is not a limitation of Claim 11.

Regarding Claim 12,

The Applicant believes that Claim 12 is allowable for at least the reasons discussed with respect to Claim 47, from which it depends.

Regarding Claim 21,

Claim 21 recites:

21. *A method for executing an operation upon a linked data structure having at least one element, the method comprising the steps of:*

- (a) *grouping a first plurality of operation tasks of the operation in a first set of operation tasks, the first set of operation tasks operable to effect a first set of element state transitions;*
- (b) *performing the first set of operation tasks in a first phase;*
- (c) *grouping a second plurality of operation tasks of the operation in a second set of operation tasks, the second set of operation tasks operable to effect a second set of element state transitions, the second set of element state transitions being distinct from the first set of element state transitions; and*
- (d) *performing the second set of operation tasks in a second phase.*

In rejecting Claim 21 the Examiner admits:

FAROOK does not explicitly detail the grouping of a first plurality of operation tasks to a first set of operations tasks to be performed. FAROOK details a new implementation developed such that certain algorithms, i.e. the cursor, insert, and delete algorithms execute a plurality of instruction operations. Therefore, it would be obvious to the teachings of FAROOK that the development of the various algorithms would require one to group the instruction operations into a set, i.e. algorithm or sub program to be run and executed ...

The Applicant traverses this statement.

First, those objects being grouped in Claim 21 are “*operation tasks*” rather than “operations.” The distinction between “*operation tasks*” and “operations,” is discussed above with respect to Claim 11. Even if one were to assume for the sake of argument that grouping of operations was well known, this would not be true of grouping of operation tasks into “*a first set of operation tasks ... operable to effect a first set of element state transitions,*” as recited in Claim 21. There does not seem to be any purpose, in the context of FAROOK, to grouping operation tasks into a specific set.

Second, Claim 21 includes two groupings: “grouping a first plurality of operation tasks” and “grouping a second plurality of operation tasks.” The Examiner has not addressed this second grouping, nor has the Examiner suggested why it would be obvious to have **two separate groupings**. As such, the Examiner has not addressed all elements of Claim 21 and has not made a prima facie case for rejection of Claim 21 under 103(a). Further, even if for the sake of argument one were to assume that grouping of “*operation tasks*” was obvious, this would not make it obvious to perform two separate grouping steps as recited in Claim 21.

Further, the Examiner does not appear to provide a motivation to include the use of a “grouping” within the teachings of FAROOK. As such, the Examiner has not presented a prima facie case for rejection of Claim 21 under 103(a). The Applicant respectfully requests that the Examiner provide a proper motivation from the prior art as required, or allow Claim 21.

For at least these reasons the Applicant believes that Claim 21 is allowable.

New Claims

Regarding Claims 41 and 48,

New Claims 41 and 48 are believed to allowable for the same reasons as Claim 30 discussed above. Claims 41 and 48 are supported by at least paragraphs 22-32, as well as
5 other places within the specification as filed.

Regarding Claim 42,

New Claim 42 is supported by at least Claim 1, as well as other places within the specification as filed.

Regarding Claim 43,

10 New Claim 43 is supported by at least Claim 5 and paragraph 24, as well as other places within the specification as filed.

Regarding Claim 44,

New Claim 44 is supported by at least Claim 36, as well as other places within the specification as filed.

15 Regarding Claim 45,

New Claim 45 is supported by at least Claim 38, as well as other places within the specification as filed.

Regarding Claim 46,

New Claim 46 is supported by at least Claim 10 and Paragraphs 22-35, as well as
20 other places within the specification as filed.

Regarding Claim 47,

New Claim 47 is supported by at least Claim 11, as well as other places within the specification as filed.

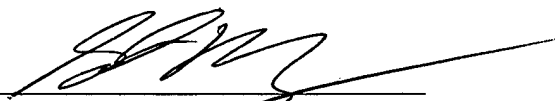
The Applicant believes that all pending claims are allowable and respectfully requests that the Examiner issue a Notice of Allowance. Should the Examiner have questions, the Applicant's undersigned representative may be reached at the number provided below.

5

Respectfully submitted,
Clifford L. Hersh

10

Date: April 12, 2006



Steven M. Colby, Reg. No. 50,250

15

Carr & Ferrell *LLP*
2200 Geng Rd.
Palo Alto, CA 94303
Phone (650) 812-3424
Fax (650) 812-3444